DEC Datasystem 500

Indent Guide

WELCOME TO INDENT
INDENT is DIGITAL's new forms management system. It is fast, powerful, and easy-to-use. This demonstration brochure introduces you to INDENT---its capabilities, features, and value to you!

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

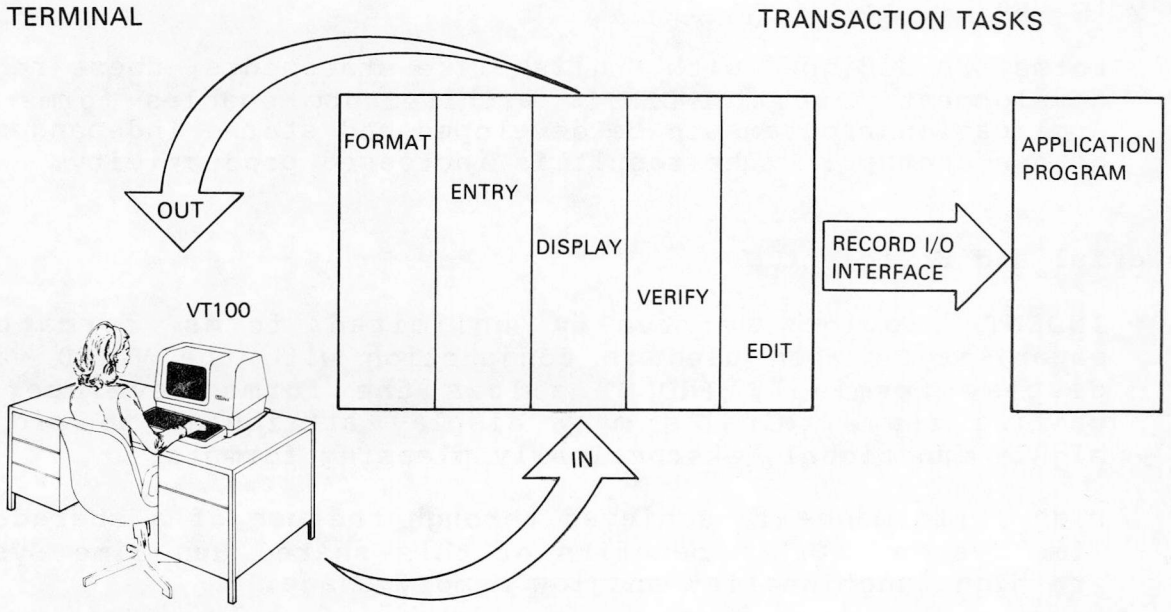
No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright 1980 by Digital Equipment Corporation

WHAT IS INDENT?

Data Entry/Validation System

Forms Management System



INDENT is a new forms management/data entry system which greatly simplifies and expedites application program development while reducing the burden on both CPU and main memory.

INDENT's powerful run-time system allows a multiterminal application to simultaneously engage in many tasks with only a single host applicaton program.

INDENT enables the form management, data entry, and data verification functions to be developed and handled independently of the application program.

INDENT employs a concise yet comprehensive command language, can be used with either the VT52 or VT100 video display terminals, and is currently supported by Digital's RSTS/E and CTS-500 operating systems.

THE BENEFITS OF INDENT

INDENT was designed to eliminate the many drawbacks associated with traditional forms management in commercial applications. It offers both user and developer benefits that result in more efficient program development and in more cost-effective application processing.

Easy to Use

Forms are designed with English-like statements; these reduce development time. INDENT's architecture enables forms and application programs to be developed and stored independently of one another. The result is increased productivity.

Powerful and Efficient

INDENT provides virtually unlimited forms formatting capability. When used in conjunction with the VT100 video display terminal, INDENT allows the forms developer to utilize the terminal's many display attributes to produce highly functional, aesthetically pleasing formats.

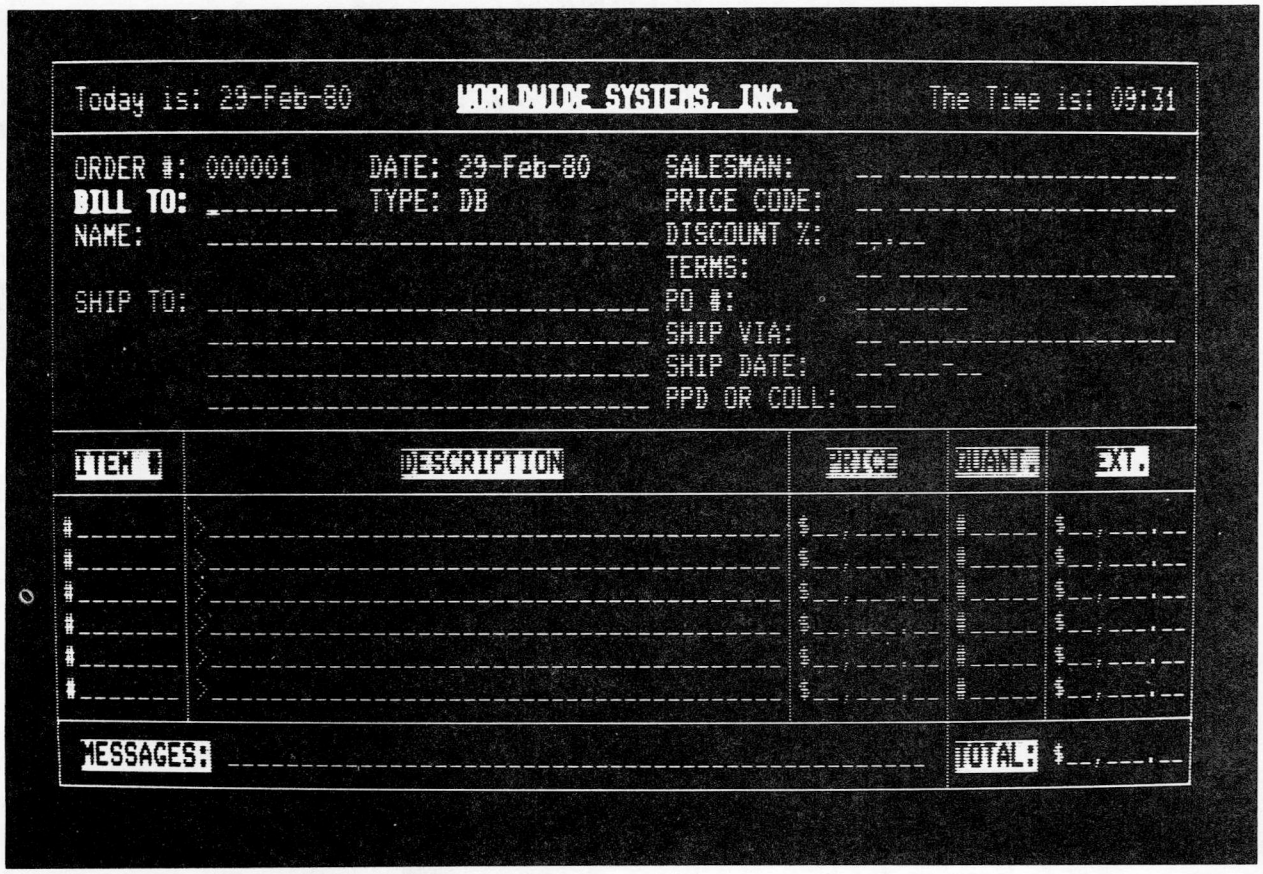
High performance is achieved through the use of a shared Run Time System. Other benefits of this shared Run Time System are high functionality and low memory usage.

Fast

Using a complex set of forms and a large, multi-file COBOL application program, a performance test of INDENT concluded that 40 terminals on an 11/70, and 16 terminals on an 11/34, all running at 3.5 characters per second input, used only 38% of the CPU. Comparable results are obtainable using DIBOL and BASIC-PLUS-2 applications. Of course, the guide for determining the number of terminals that INDENT will support is dependent upon the operating system configuration, the specific application program being used, and other concurrent system activity.

Multiple Screen Options

INDENT offers very flexible screen handling. Forms can be displayed all at once or one field at a time. Fields from different forms can be displayed one after another. Fields from a single form can be displayed in an order different from that in which they occur in the form. Forms can be chained together. Portions of a form can be scrolled on the screen. Several forms can be displayed simultaneously. These options provide users with almost unlimited flexibility for managing terminal-intensive applications.



Flexible Data Handling

INDENT validates all data input and output from the operator. This means that data integrity is assured without the need to implement applicaiton routines for verification, or to make preprocessing edit runs on newly entered data. INDENT allows data to be restricted to a certain range of values or to a list of specific values. Data fields can be justified right or left. Within a given data field, INDENT allows different characters to be specified as alphabetic, digital, numeric, alphanumeric, or printable ASCII. It allows numeric data to be zoned, overpunched, leading separate sign, or packed.

Error Handling Capability

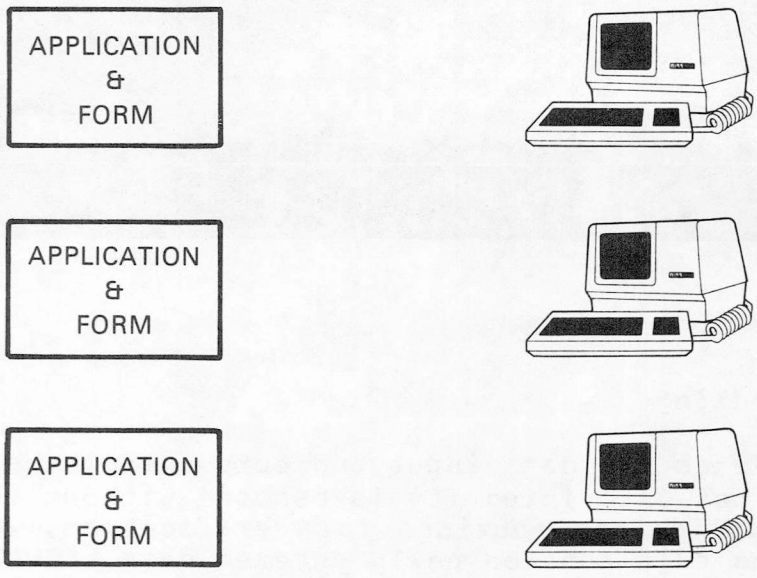
INDENT provides complete error handling capability. Not only are the error messages self-explanatory, but they can be overridden for a given data field. Because of these features, terminal operators require less training and can work with greater confidence.

WHY INDENT?

Application programs have traditionally included the application coding as well as the code for forms, data entry, and data verification. All of this code was written in a high-level language.

This traditional approach required considerable coding and often produced programs which were long and which made heavy demands on the CPU and main memory. Changes to forms required overall changes to the program. Multiterminal applications required multiple copies of each program.

TRADITIONAL

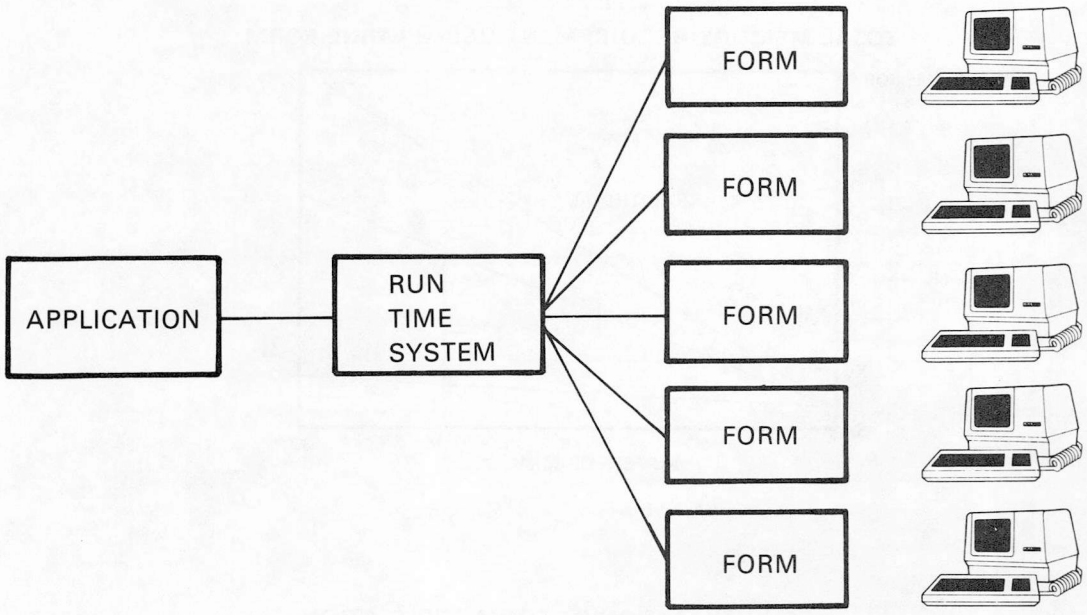


The traditional approach to terminal-intensive application processing requires a large terminal handler plus a copy of the entire forms management/application program for each terminal working on the same task. This places an excessive burden on the system and degrades terminal response.

INDENT is more efficient than the traditional approach. This efficiency reduces the burden on the CPU and main memory, and allows the host system to provide better terminal response to more terminals.

Application programming is simpler and faster. Changes to host programs, data management, or forms definitions can be implemented independently of one another. Because the **INDENT** command language is so easy to use, an entry level programmer can quickly learn to create and modify forms definitions.

INDENT



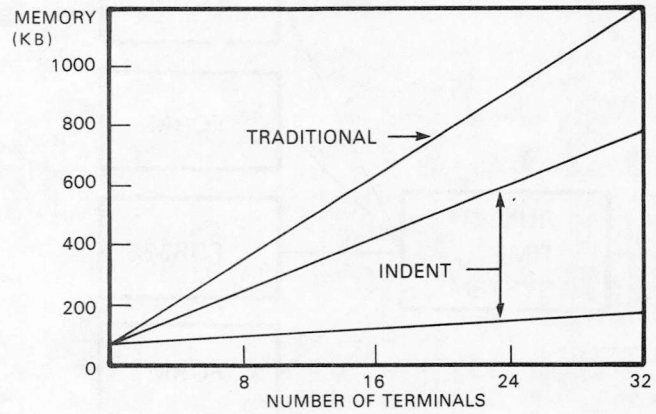
INDENT utilizes a powerful run-time system which enables the forms management, data entry, and data verification coding to be separate from the application programs. A single copy of the application program can serve all terminals performing the same task. This greatly reduces the burden on CPU and main memory, and permits a high level of terminal responsiveness.

INDENT MEANS PERFORMANCE

The power of INDENT is three-fold: (1) it greatly reduces the burden on CPU and main memory, (2) it greatly simplifies and expedites the programming effort, and (3) it provides terminal operators with a powerful set of keypad commands for application processing.

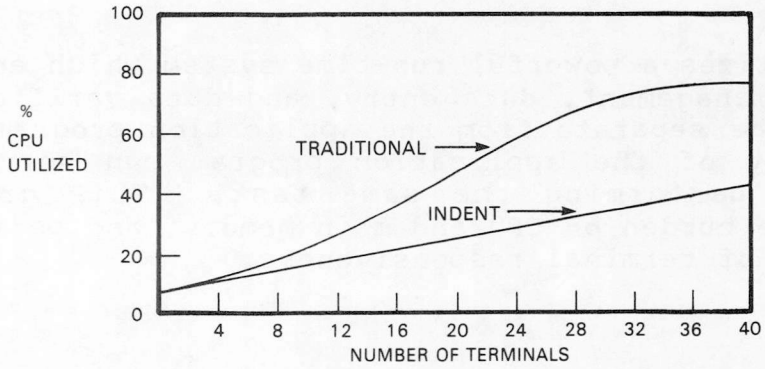
The following chart illustrates INDENT performance using a complex set of forms and a large, multi-file COBOL application program. The bottom line shows multiple terminals running the same application and the middle line represents each terminal running a different application. In either case, memory savings is substantial compared to the traditional approach (top line), where multiple terminals are running one application.

TOTAL MEMORY REQUIREMENT USING LARGE FORM



CPU UTILIZATION OF FORM APPLICATION

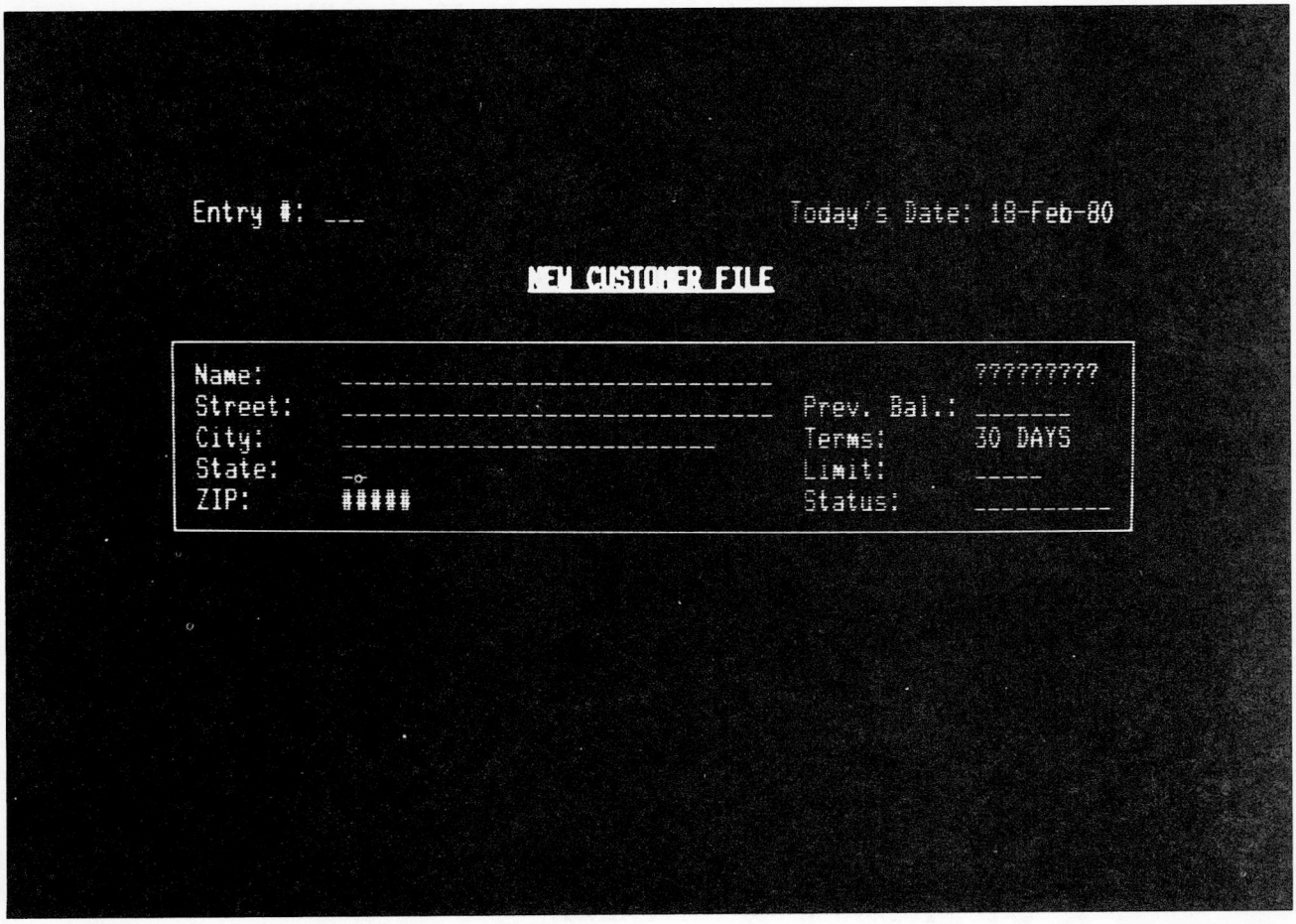
[11/70 RSTS V7.0]



The chart above illustrates INDENT's minimal impact on CPU utilization as the number of terminals increases. The traditional approach consumes far more processor power for the same increase in number of users.

INDENT MEANS EASE OF USE

Below is a typical data entry form as it appears on the display screen for a terminal operator. Following it are (1) the INDENT form definition which generates the form, (2) a COBOL application program which uses this form, (3) a DIBOL applicaton which uses this form, and (4) a BASIC-PLUS-2 application which uses this form. Note how the INDENT forms definition is coded; the three application programs contain no forms coding.



Screen Display of an INDENT Form

```
BOX       SQR:(9-15,6-73)
TABLE     WORDS: 'GOLDEN', 'OK', 'DELINQUENT', 'NEVERMORE'

FORM      DATENT,          ERASE, CUE, BOX:SQR

DEFAULT   HALT

,         (5,8)            'Entry #: ',          n3,    INC, PRO
,         (+0,+41)         "Today's Date: ",     A7,    DAT, PIC:'XX-XXX-XX',
&                                                        PRO

,         (7,31)           'NEW CUSTOMER FILE',         UND, BOL

NAM,      (+3,8,18)        'Name:',              A30
ACNTNO,   (+0,50,62)       'Accnt. No.:',        N9,    REV:P, PAI:'?',ZER
,         (11,8,18)        'Street:',            A30
,                          'City:'               A26
,                          'State:',             A2,    -HAL
,                          'ZIP:',               N5,    ZER, PAI:'#'

,         (11,50,62)       'Prev. Bal.:',        N7,    PRO
,                          'Terms:',             A8,    SKI, INI:'30 DAYS'
,                          'Limit:',             N5,    UPP:20000
,                          'Status:',            A10,   LOO:WORDS
```

INDENT Program which Generates the Form

```
IDENTIFICATION DIVISION.
PROGRAM-ID. DEMO.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT CUSTOM  ASSIGN TO "CUSTOM.DAT" ORGANIZATION IS INDEXED
                  RECORD KEY IS ACCNT.
DATA DIVISION.
FILE SECTION.
FD  CUSTOM LABEL RECORD IS STANDARD; VALUE OF ID IS "CUSTOM.DAT".
01  DATEN.  02 F1 PIC X(40). 02 ACCNT PIC X(9). 02 F2 PIC X(93).
WORKING-STORAGE SECTION.
77      DIAG    PIC     99; USAGE IS COMP; VALUE 0.
77      LIT-1   PIC     9; USAGE IS COMP; VALUE IS 1.
77      LIT-KB  PIC     X(4); VALUE IS "KB: ".
77      LIT-FORM1 PIC   X(6); VALUE IS "DATENT ".
PROCEDURE DIVISION.
BEGIN.  OPEN OUTPUT CUSTOM.
        CALL "FDIAG" USING LIT-1,DIAG.
        CALL "FSTRT" USING LIT-1,LIT-KB,LIT-FORM1.
AGAIN.  CALL "FINPT" USING LIT-1,LIT-FORM1,DATEN.
        IF FLAG = -1 GO TO STOP1.
        WRITE INBUF; INVALID KEY GO TO STOP1.  GO TO AGAIN.
STOP1.  CALL "FSTOP" USING LIT-1.  CLOSE CUSTOM. STOP RUN.
```

COBOL Program which uses the INDENT Form

```
START
RECORD DATEN
       FILL1,  A40
       ACCNT,  D9
       FILL2,  A93
RECORD WORK
       DIAG,   D3
PROC
INITS, OPEN (1,0,'CUSTOM.DAT')
       XCALL FDIAG (1,DIAG)
       XCALL FSTRT (1,'KB:','DATENT')
LOOP,  XCALL FINPT (1,'DATENT',DATEN)
       IF (DIAG.EQ.-1) GOTO ENDIT
       WRITES (1,DATEN)
       GOTO LOOP
ENDIT, XCALL FSTOP (1)
       CLOSE (1)
STOP
```

DIBOL Program which uses the INDENT Form

```
900       MAP (MAP1)    FIL1$=40%, ACCNT$=9%, FILL2$=93%      &
        \ MAP (MAP1)    DATEN$ = 140%
910       OPEN 'CUSTOM.DAT' AS FILE #1%,                       &
                        ORGANIZATION INDEXED,ACCESS MODIFY,    &
                        ALLOW MODIFY,MAP MAP1,PRIMARY ACCNT$   &

1000      CALL FDIAG    BY REF (1%, DIAG%)
2000      CALL FSTRT    BY REF (1%,'KB: ','DATENT ')
3000      CALL FINPUT   BY REF (1%,'DATENT ',DATEN$)           &

3100      IF D% = -1% GOTO 29000
4000      PUT #1%
5000      GOTO 3000                                            &

29000     CALL FSTOP    BY REF (1%)                            &

29100     CLOSE #1%
32767     END
```

BASIC-PLUS-2 Program which uses the INDENT Form

INDENT MEANS FUNCTIONALITY

INDENT's form language elements, host program commands, terminal keypad functions, and efficient directives allow a programmer to create an almost unlimited number of forms for application programming.

POWERFUL FORM LANGUAGE ELEMENTS

CUE . . . . Bolds prompts for fields accepting input.
SKIP. . . . Skips over fields when logically reached.
PICTURE . . Formats and validates fields as pictures.
INCREMENT . Increments values in fields by one.
INITIAL . . Sets initial values for fields.
DUPLICATE . Displays previous field values during input.
PAINT . . . Displays specified character at start of input.
LOOKUP. . . Performs table lookup on specified table.
SUBSTITUTE. On a match, substitutes table 1 value for table 2 value.
REQUIRED. . Requires data to be entered before field exit.
LOWER . . . Sets lower limit range check (also UPPER).
COPY. . . . Copies verified data into specified field.

And many more!

FLEXIBLE TERMINAL KEYPAD FUNCTIONS

ADVANCE and BACK FIELD . . . For rapid data entry and editing.
ADVANCE and BACK FORM. . . . For displaying chained forms.
HELP, OVERRIDE and ABORT . . For operator assistance and form control.
DELETE FIELD and CANCEL. . . For editing and keypad control.

And more.

INDENT's rich set of form definition attributes provides form designers with the flexibility they need to get the job done. Many attributes can be applied to both forms and fields; others are specifically intended for form or field control. Multiple lines of attributes can be applied to fields for comprehensive field definition.

EFFICIENT DIRECTIVES FOR FULL FORM CONTROL

TABLE. . . Defines table list (literals AND variables).
BOX. . . . Defines boxes and lines.
DEFAULT. . Sets user-defined defaults.
BLOCK. . . Defines variables that are loadable from host programs.

POWERFUL HOST PROGRAM COMMANDS

FSTRT. . . Starts form file tasks.
FDIAG. . . Provides host program with complete diagnostics.
FSHOW. . . Displays forms or fields.
FINIT. . . Displays forms or fields with initial values.
FINPT. . . Accepts terminal input from forms or fields.
FSET . . . Displays forms or fields with previously entered values.
FGET . . . Retrieves current data from forms or fields.
FPUT . . . Puts data to forms or fields.
FABRT. . . Terminates execution of current requests.
FSTOP. . . Stops form tasks.

And many more including a full set of multiterminal commands!

Functionality is further provided through INDENT's powerful set of forms directives and host program commands. Boxes and lines can be quickly defined; tables can include lists of literals or variables; form variables can be defined and accessed by host programs; forms can initialize host programs and host programs can initialize forms; defaults can be set and reset.

INDENT MEANS BUSINESS

OEMs

INDENT reduces development costs. The same application program can be adapted to different customers by using different INDENT form definitions. The same INDENT form can be used for different applications to increase uniformity across application packages.

PROGRAMMERS

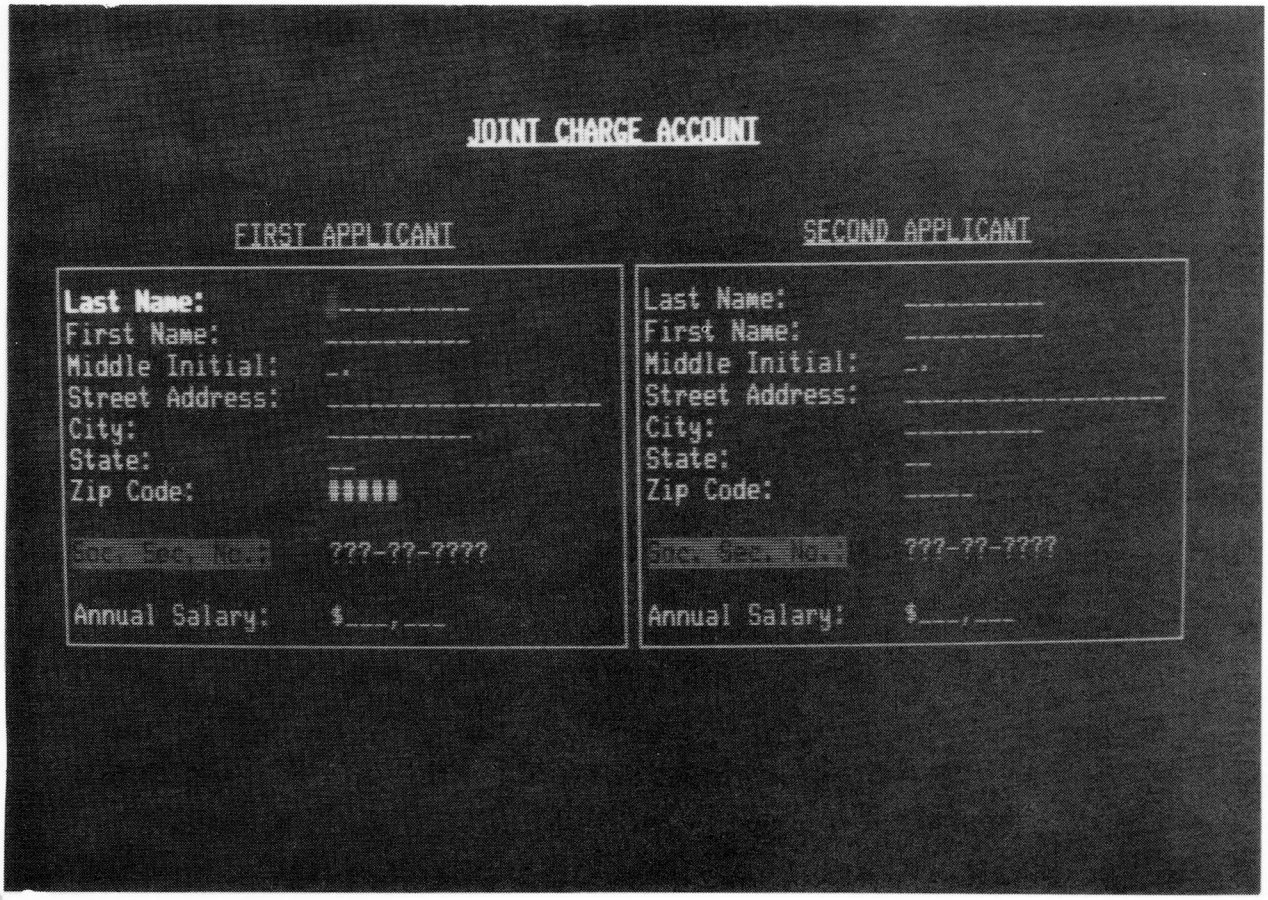
INDENT reduces design and coding time. Application programs are easily interfaced with INDENT. INDENT validates and structures data; data is as clean as a record read from a disk file.

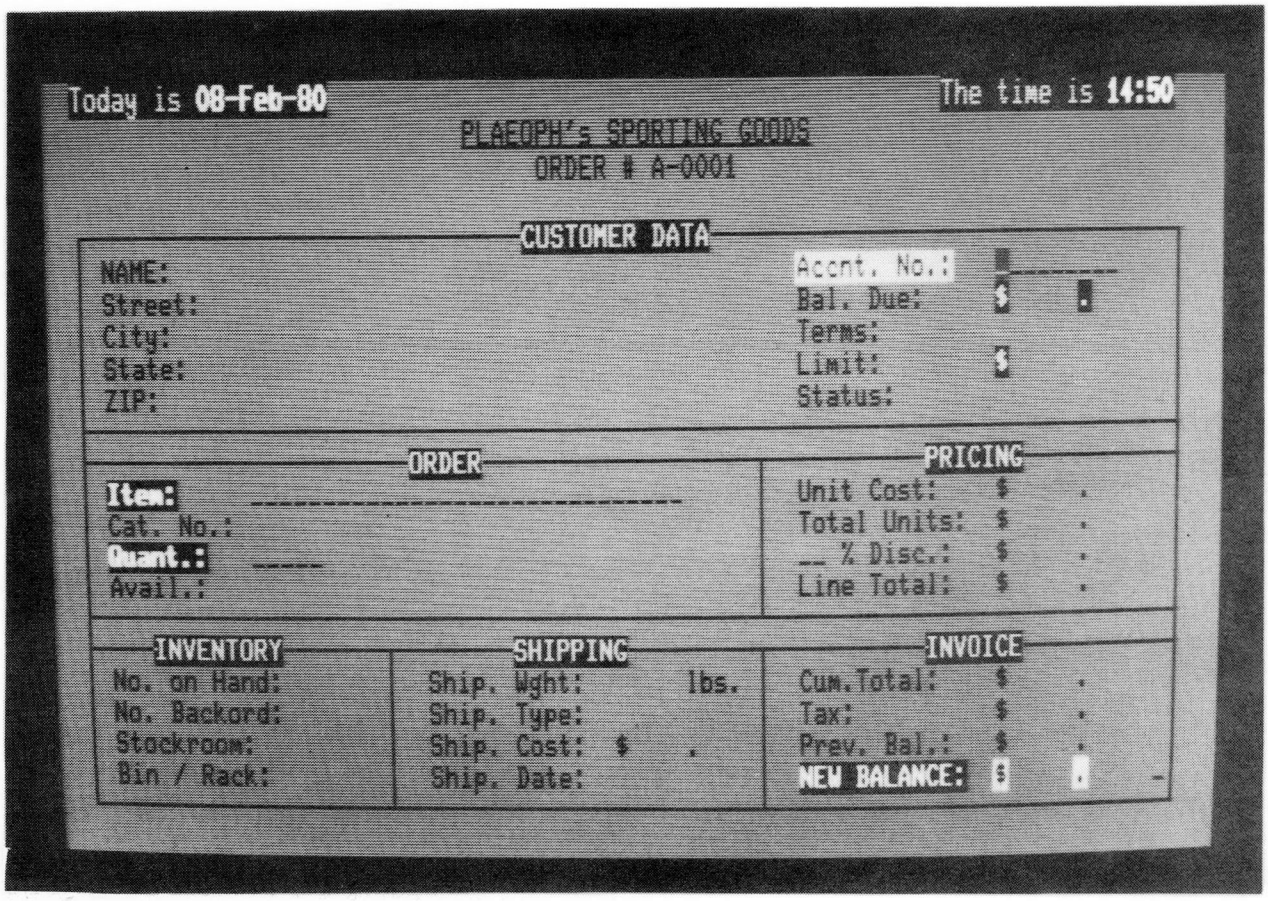
FORM DESIGNERS

INDENT reduces programming costs. The field definitions are simple so the forms designers don't have to be programmers. Designers don't have to worry about the host program.

DATA-ENTRY CLERKS

The degree of operator control is predetermined. The keypad is simple and versatile; the terminal responds very quickly. Field-by-field entries are prompted. The formatting and sequencing of forms and fields is automatic. Validation is automatic.


JOINT CHARGE ACCOUNT
FIRST APPLICANT
SECOND APPLICANT
Last Name:
First Name:
Middle Initial:
Street Address:
City:
State:
Zip Code:
Soc. Sec. No.:
Annual Salary:
Last Name:
First Name:
Middle Initial:
Street Address:
City:
State:
Zip Code:
Soc. Sec. No.:
Annual Salary:


Today is 08-Feb-80
The time is 14:50
PLAEOPH's SPORTING GOODS
ORDER # A-0001
CUSTOMER DATA
NAME:
Street:
City:
State:
ZIP:
Accnt. No.:
Bal. Due:
Terms:
Limit:
Status:
ORDER
Item:
Cat. No.:
Quant.:
Avail.:
PRICING
Unit Cost:
Total Units:
% Disc.:
Line Total:
INVENTORY
No. on Hand:
No. Backord:
Stockroom:
Bin / Rack:
SHIPPING
Ship. Wght:
lbs.
Ship. Type:
Ship. Cost:
Ship. Date:
INVOICE
Cum.Total:
Tax:
Prev. Bal.:
NEW BALANCE:

PRINTED IN USA ED 18996 80/123 05 12 3.5

digital